ESC-TR-99-057

Technical Report
1060

Bias Removal Techniques for the Target-Object Mapping Problem

C.J. Humke

9 July 2002

Lincoln Laboratory
MASSACHUSETTS INSTITUTE OF TECHNOLOGY
LEXINGTON, MASSACHUSETTS

Prepared for the Department of Defense, MDA, under Air Force Contract F19628-00-C-0002.

This report is based on studies performed at Lincoln Laboratory, a center for research operated by Massachusetts Institute of Technology. This work was sponsored by the Department of Defense, Missile Defense Agency, under Air Force Contract F19628-00-C-0002.

This technical report has been reviewed and is approved for publication.

FOR THE COMMANDER

Gary Tutungian
Administrative Contracting Officer
Contracted Support Management

Massachusetts Institute of Technology
Lincoln Laboratory

Bias Removal Techniques for the Target-Object Mapping Problem

C.J. Humke
Group 32

TECHNICAL REPORT 1060

9 July 2002

Lexington Massachusetts

ABSTRACT

Multisensor track and data fusion require accurate sensor registration and positional bias removal. This report presents the problem of bias removal in the context of midcourse interceptor engagements. For this application two sensors are employed: a fire-control radar and an interceptor with an optical seeker. The fire-control radar tracks objects, collects data on the objects, performs discrimination, and hands over a target object map to the interceptor. Positional bias is generally evident in the target object map and must be removed in order to fuse the radar and seeker data accurately. Equations and algorithms are derived for calculating the maximum-likelihood bias estimate for three scenarios, given three assumptions about the measurement errors on the radar- and seeker-tracked objects. First, bias is shown to be an insignificant contributor to the assignment of objects when both sensors track the same objects and the measurement errors are equal and uncorrelated for all objects. Second, equations for direct calculation of the optimal bias are derived for cases where one sensor tracks a subset of the objects tracked by the other sensor. Third, algorithms are discussed for the most complicated scenarios, those where dissimilar objects are tracked by the two sensors.

ACKNOWLEDGMENTS

The author acknowledges the support and guidance provided by Dr. R. Louis Bellaire, Dr. Daniel A. O'Connor, and Dr. Keh-Ping Dunn. Louis Bellaire has offered insights that have greatly contributed to this work, as well as suggestions that have improved the readability and accuracy of this report. Dan O'Connor and Keh-Ping Dunn have been key motivators of this work and of this report.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF TABLES

1. INTRODUCTION

In most ballistic missile defense scenarios, the reentry vehicle (which carries the warhead) will be accompanied by associated objects such as tanks and deployment debris. The interceptor (which has an optical seeker) detects some of these objects on its focal plane and engages the most threatening. Several sources of information may be employed to correctly identify the reentry vehicle as the most threatening object in the complex. In principle, discrimination may be accomplished by (1) the fire-control radar, which then hands over information to the interceptor; (2) the interceptor seeker autonomously, using its own measurements; or (3) the radar and seeker together, by combining data for fused discrimination. Regardless of the discrimination architecture selected, handover of radar data to the interceptor is meaningful only if the objects seen by the radar and those seen by the seeker are correctly associated with one another. This report considers the handover of radar data necessary to the interceptor and proposes robust algorithms for the association of objects between the radar and the seeker. Figure 1 is a cartoon of the detection and handover process.

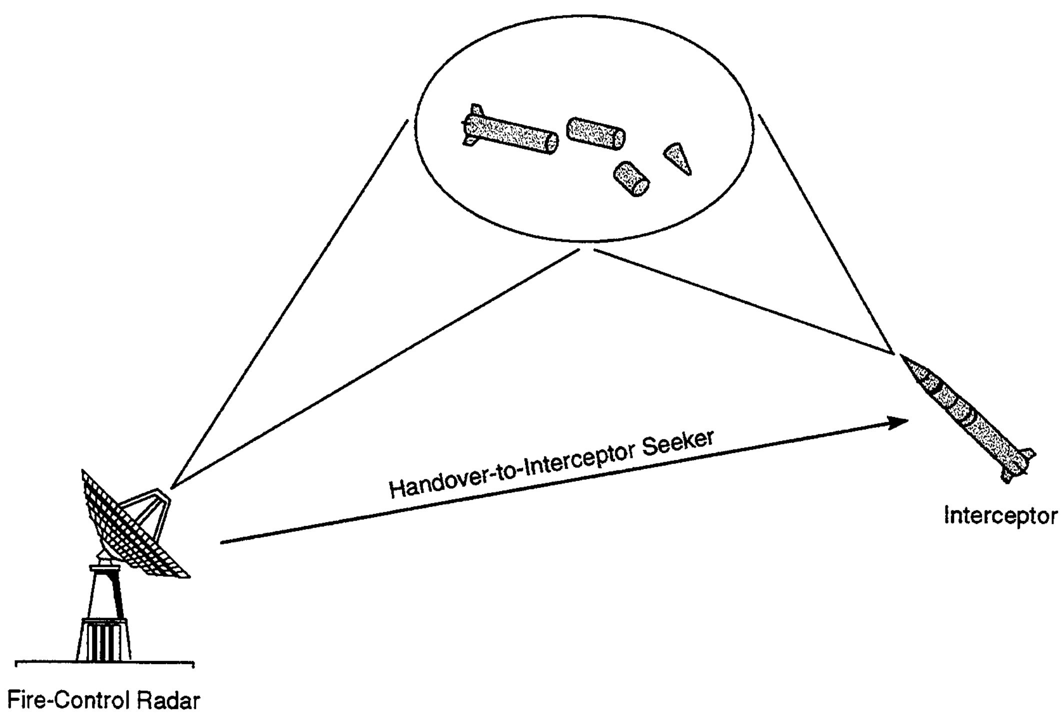

Figure 1. Handover of tracked objects from the radar to the interceptor.

To ensure a successful handover, the information contained in the radar uplink message must be sufficient to unambiguously associate the reentry vehicle. A number of signature and metric features are used for this association, including angle, angle rate, target size, and scintillation. The focus of this report is on one of the most commonly used correlates, focal-plane position. The positions of the radar objects are mapped to the seeker focal plane as azimuth and elevation positions and then matched with the positions of the seeker objects. This process is referred to as a "target-object map" (TOM). The TOM may be modeled as a linear-programming problem and solved using a minimum-weight matching algorithm such as Munkres [1,2] or JVC [3,4]. The linear-programming problem searches for a set of variables that minimize a linear cost function by taking into account linear equality and/or inequality constraints [5].

The generic assignment problem is shown in Figure 2, where the objective is to assign an object from one source to an object from another. The assignment problem is a special subset of linear-programming problems because it has the constraint that all assignments are mutually exclusive and collectively exhaustive. An example is the assignment of a person to a task, where each person would be assigned once and only once to a task and each task would be assigned once and only once to a person. The constraints shown in Figure 2 ensure that each column and each row of the decision matrix D [where $D(i,j) = d_{ij}$] is assigned once and only once and that the decision variables are binary.

Objective: minimize $z = \sum_i \sum_j c_{ij} d_{ij}$

subject to: $\sum_i d_{ij} = 1 \quad \forall\, j$

$\sum_j d_{ij} = 1 \quad \forall\, i$

$d_{ij} = [0 \text{ or } 1] \quad \forall\, i, j$

where d_{ij} represents the decision to assign (1) or not assign (0) object i from one source to object j from another source, and c_{ij} is the cost of the decision d_{ij}.

Figure 2. Generic assignment problem.

In the TOM problem the costs c_{ij} are derived from a model of the relative metric errors between the two sensors. A few types of errors can occur during the uplink process that add a translational bias to the uplinked focal-plane positions of the radar objects and to the positions of the seeker objects. Random, zero-mean, white radar track and seeker measurement errors further complicate the association problem. Registration bias is illustrated on the left-hand side of Figure 3; random errors that result from measurement noise are shown on the right-hand side. Measurement noise errors cannot be removed; however, bias can be removed by shifting the uplinked radar objects to minimize the distance between the radar and seeker objects. This optimal shift will be the maximum-likelihood bias estimate and will be embedded in the costs c_{ij} shown in Figure 2.

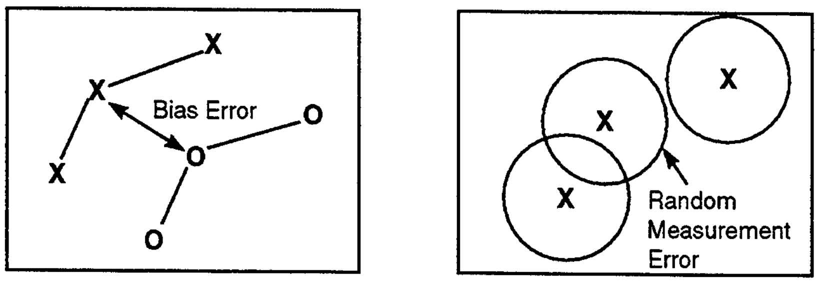

Figure 3. Bias error, random error.

The objective of this report is to propose algorithms for calculating optimal bias. The problem is divided into three possible scenarios. The first, the m x n scenario, is defined with m objects in common between the two sensors. The second scenario, m x n, is defined such that m objects are in common between the two sensors and one sensor has $n - m$ additional objects. The third scenario, the mismatch, assumes that at least one object is common and at least one object is different between the two sensors. Venn diagrams of each scenario are shown in Figure 4. The object set for one sensor is depicted with horizontal stripes and the set for the other sensor has vertical stripes, such that the set of objects in common is crosshatched.

The rest of this report is organized as follows: Section 2 defines the assignment problem and develops the cost-function model. Section 3 develops four proposed algorithms for calculating bias for the m x m and m x n scenarios. Finally, Section 4 offers a summary and identifies issues for future work. As the algorithms and discussions in this report are applicable to problems beyond the radar-to-seeker TOM, the remainder of the material will address the problem generically. The reader is encouraged to draw parallels with other two-sensor tracking correlation problems where registration biases play an important role.

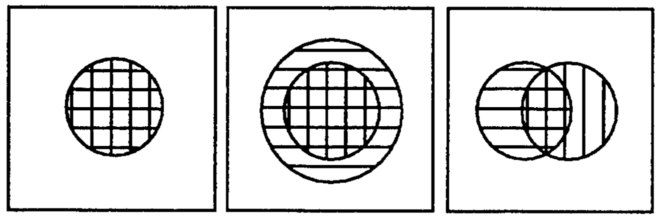

Figure 4. Venn diagrams of scenarios.

2. STATEMENT OF PROBLEM

The objective of the track-to-track assignment problem is to find the optimal minimum error assignment of objects viewed by two sensors. This problem is complicated by the presence of registration biases that shift the positions of all objects from truth. The total bias may have an effect on the assignment; therefore, bias must be accounted for to ensure the accuracy of the track-to-track assignment. The generic assignment problem shown in Figure 2 will now be tailored using the particular features that distinguish the track-to-track assignment problem, such as bias and the number of common objects.

Bias is defined as the shifted difference in position between the objects of the two sensors. Let (x_i, y_i) be the azimuth and elevation positions, respectively, of the *i*th object for sensor one; let (u_j, v_j) be the azimuth and elevation positions, respectively, of the *j*th object for sensor two. If there are no random measurement errors, bias $[b_u\ b_v]$, affects all targets equally so that $u_j = x_i - b_u$ and $v_j = y_i - b_v$ when $d_{ij} = 1$. Given the presence of random errors, the optimal assignment may be calculated by weighting the positional differences. When the errors are Gaussian, the weights are the measurement error covariance matrices for the *i*th and *j*th objects of sensors one and two, P_i and Σ_j, respectively. The cost function from Figure 2, rewritten to include the assumed error model, closely resembles a linear least-squares cost as shown in Figure 5. The cost function in Figure 5 minimizes the log likelihood, as derived in the Appendix.

$$c_{ij} = \ln\left(\left|P_i\right|\right) + \ln\left(\left|\Sigma_j\right|\right) + \begin{bmatrix} x_i - b_u - u_j \\ y_i - b_v - v_j \end{bmatrix}^T \left(P_i + \Sigma_j\right)^{-1} \begin{bmatrix} x_i - b_u - u_j \\ y_i - b_v - v_j \end{bmatrix}$$

where $[x_i\ \ y_i]$ is the position of the *i*th object for sensor one,

$[u_j\ \ v_j]$ is the position of the *j*th object for sensor two,

$[b_u\ \ b_v]$ is the bias,

P_i is the covariance matrix of the *i*th object for sensor one, and

Σ_j is the covariance matrix of the *j*th object for sensor two.

Figure 5. Weighted-distance cost function.

Given the presence of bias, the problem is nonlinear programming with respect to b_u, b_v, and (d_{ij}) instead of a linear problem of (d_{ij}) alone (as in Figure 2). Ideally, an attempt would be made to solve the problem presented in Figure 5 directly, solving for bias and decision variables at the same time; however, it is difficult and computationally expensive to solve a nonlinear binary problem accurately. Alternative algorithms, therefore, have been developed that calculate bias before solving for decision variables; these alternative algorithms are presented in this report.

The constraints (see Figure 6) also change from those shown in Figure 2 to allow for an unequal number of objects between the two sensors. (Note that if both sensors track the same objects, the constraints in Figure 6 are the same as those in Figure 2.) Without loss of generality, it is assumed in this report that $m \leq n$; it is also assumed that each sensor contributes all the objects that it is able to track to the track-to-track assignment problem.

Assumptions: (1) sensor one tracks m objects

(2) sensor two tracks n objects, where

(a) objects $[1:m]$ are the same as the m objects for sensor one

(b) objects $[m+1:n]$ are extra objects

Constraints:

$$\sum_i d_{ij} = 1 \quad \forall\, j \in [1:m]$$

$$\sum_i d_{ij} = 0 \quad \forall\, j \in [m+1:n]$$

$$\sum_j d_{ij} = 1 \quad \forall\, i \in [1:m]$$

Figure 6. Constraints adjusted for an unequal number of objects.

Although the most general form of the bias calculation problem can be complex and computationally expensive to solve, assumptions can be made to constrain the problem and improve its solvability. These special cases provide valuable insight into the track-to-track assignment problem. For each of the three scenarios, m x m, m x n, and mismatch, one of three assumptions may be made about the error covariances of the objects for a given sensor. As mapped in Table 1, five different bias calculation algorithms have been developed to handle these three scenarios, as well as the three assumptions. First, it may be assumed that the covariance matrices for a given pair of objects sum either to the identity matrix, $P_i + \Sigma_j = I$, or to a diagonal matrix, $P_i + \Sigma_j = cI$, where c is any vector $\left[\sigma_{az}^2 \;\; \sigma_{el}^2\right]$. (Note that cost function is reduced to a Euclidean-distance function when $P_i + \Sigma_j = I$.) Second, it may be assumed that the covariance matrices for each sensor are uncorrelated and identical for all objects such that $P_i + \Sigma_j = W$, where W denotes a constant matrix that is independent of i or j. Third, if no assumptions are made, the covariance matrices are modeled as unique for each object such that $P_i + \Sigma_j = W_{ij}$, where W_{ij} denotes a matrix that is dependent on i and j. The remaining sections of this report present algorithms for calculating bias for these three scenarios and these three assumptions. Although real-time algorithms will not be able to use scenario typing, stepping through these algorithms for special cases can assist in the formulation, albeit suboptimal, of bias removal algorithms for the general problem.

TABLE 1

Map of Assumptions, Scenarios, and Proposed Algorithms

Covariance Assumption	Scenario Assumption	Case	Bias Algorithm*
$P_i + \Sigma_j = I$ $P_i + \Sigma_j = cI$	$m \times m$ $m \times n$ mismatch	1 3 5	Insignificant Loop through, Equation (10) Loop through, Equation (7)
$P_i + \Sigma_j = W$	$m \times m$ $m \times n$ mismatch	2 3 5	Direct estimation Equation (8) Loop through, Equation (10) Loop through, Equation (7)
$P_i + \Sigma_j = W_{ij}$	$m \times m$ $m \times n$ mismatch	4 4 5	Loop through, Equation (6) Loop through, Equation (6) Loop through, Equation (6)
*Equations are shown in Section 3.			

3. BIAS CALCULATION ALGORITHMS

3.1 CASE 1: BIAS AS AN INSIGNIFICANT CONTRIBUTOR

This first case is presented in order to understand the effect of bias on the optimal assignment for the m x m scenario when $P_i + \Sigma_j = cI$. It will be shown that bias may be analytically factored from cost function, thereby making bias an insignificant contributor to the optimal decision. To understand how bias affects cost function, the objective function from Figure 5 is first manipulated. (The problem is worked through using only scalar positions in order to preserve the simplicity of the mathematics.)

$$
\begin{aligned}
z &= \frac{1}{c_1}\sum_i\sum_j\left(x_i - u_j - b_u\right)^2 d_{ij} \\
&= \frac{1}{c_1}\sum_i\sum_j\left(x_i^2 + u_j^2 + b_u^2 - 2x_iu_j - 2x_ib_u + 2u_jb_u\right)d_{ij} \\
&= \frac{1}{c_1}\left(\sum_i x_i^2\sum_j d_{ij} + \sum_j u_j^2\sum_i d_{ij} + b_u^2\sum_i\sum_j d_{ij} - 2\sum_i\sum_j x_iu_jd_{ij} - 2b_u\sum_i x_i\sum_j d_{ij}\right) \ .
\end{aligned} \tag{1}
$$

The following constraints for an m x m problem are assumed:

$$\sum_i d_{ij} = 1 \text{ and } \sum_j d_{ij} = 1, \text{ for each } i \text{ and } j, \text{ and } \sum_i\sum_j d_{ij} = m \ . \tag{2}$$

Inserting the constraints from Equation (2) into (1) gives the following objective function:

$$z = \frac{1}{c_1}\left(\sum_i x_i^2 + \sum_j u_j^2 + b_u^2 m - 2\sum_i\sum_j x_iu_jd_{ij} - 2b_u\sum_i x_i - 2b_u\sum_j u_j\right) \ . \tag{3}$$

The objective in the assignment problem is to select values for decision variables such that cost function is minimized. If a term in the cost function does not depend on the decision variables, its contribution to the cost of the decision is that of a constant; that is, it does not encourage or discourage a given selection of the decision variables. Employing this observation simplifies the objective function in Equation (3) to z', where $z' = k_1 + k_2 z$, and k_1 and k_2 are constants, as follows:

$$
\begin{aligned}
&\frac{1}{c_1}\left(\sum_i x_i^2 + \sum_j u_j^2 + b_u^2 m - 2b_u \sum_i x_i - 2b_u \sum_j u_j\right) = \text{ constant} \\
&z' = \frac{c_1}{2}(z - \text{constant}) = -\sum_i \sum_j x_i u_j d_{ij}
\end{aligned}
\tag{4}
$$

Bias does not appear in the new objective function in Equation (4) and therefore does not influence the solution. The optimal assignment of objects is independent of bias in the m x m scenario, where both sensors track the same objects and the measurements errors are uncorrelated $\left(P_i + \Sigma_j = cI\right)$. The algorithm for this first case thus requires no special treatment of bias prior to applying one of the classic assignment algorithm solvers such as Munkres or JVC.

3.2 GENERAL SOLUTION OF BIAS

Although no bias removal is necessary for the m x m scenario when $P_i + \Sigma_j = cI$, it is necessary in all other cases that are to be considered. A general equation for calculating bias is derived and discussed in this section and is referenced in the remaining sections. To solve for the bias shown in Figure 5, the objective function is manipulated as follows:

$$
\begin{aligned}
&z = \sum_i^m \sum_j^n \left(\ln(|P_i|) + \ln(|\Sigma_j|) + \begin{bmatrix} x_i - u_j - b_u \\ y_i - v_j - b_v \end{bmatrix}^T \left(P_i + \Sigma_j\right)^{-1} \begin{bmatrix} x_i - u_j - b_u \\ y_i - v_j - b_v \end{bmatrix} \right) d_{ij} \\
&\frac{\partial z}{\partial b} = \sum_i^m \sum_j^n \frac{\partial}{\partial b}\left(\ln(|P_i|) d_{ij}\right) + \sum_i^m \sum_j^n \frac{\partial}{\partial b}\left(\ln(|\Sigma_j|) d_{ij}\right) + \sum_i^m \sum_j^n \frac{\partial}{\partial b}\left(\begin{bmatrix} x_i - u_j - b_u \\ y_i - v_j - b_v \end{bmatrix}^T \left(P_i + \Sigma_j\right)^{-1} \begin{bmatrix} x_i - u_j - b_u \\ y_i - v_j - b_v \end{bmatrix} d_{ij} \right) \\
&\frac{\partial z}{\partial b} = -2 \sum_i^m \sum_j^n \left(P_i + \Sigma_j\right)^{-1} \begin{bmatrix} x_i - u_j - b_u \\ y_i - v_j - b_v \end{bmatrix} d_{ij} \quad .
\end{aligned}
\tag{5}
$$

The equation for bias may be formed by setting $\partial z / \partial b = 0$ and then solving for b:

$$\frac{\partial z}{\partial b} = -2\left(\sum_i^m\sum_j^n\left(P_i+\Sigma_j\right)^{-1}\begin{bmatrix} x_i-u_j \\ y_i-v_j \end{bmatrix} d_{ij} - \sum_i^m\sum_j^n\left(P_i+\Sigma_j\right)^{-1}\begin{bmatrix} b_u \\ b_v \end{bmatrix} d_{ij}\right) = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

$$\left(\sum_i^m\sum_j^n\left(P_i+\Sigma_j\right)^{-1} d_{ij}\right)\begin{bmatrix} b_u \\ b_v \end{bmatrix} = \sum_i^m\sum_j^n\left(P_i+\Sigma_j\right)^{-1}\begin{bmatrix} x_i-u_j \\ y_i-v_j \end{bmatrix} d_{ij} \tag{6}$$

$$\begin{bmatrix} b_u \\ b_v \end{bmatrix} = \left(\sum_i^m\sum_j^n\left(P_i+\Sigma_j\right)^{-1} d_{ij}\right)^{-1}\left(\sum_i^m\sum_j^n\left(P_i+\Sigma_j\right)^{-1}\begin{bmatrix} x_i-u_j \\ y_i-v_j \end{bmatrix} d_{ij}\right) .$$

The general solution for the maximum-likelihood bias estimate is shown in Equation (6). Bias cannot be calculated directly using Equation (6), as the unknown decision variables (d_{ij}) are embedded. The error covariance assumptions mapped out in Table 1 are employed in the remaining sections of this report to manipulate Equation (6) to be a more-applicable estimator.

3.3 CASE 2: DIRECT CALCULATION OF BIAS

Case 2 is defined by the assumption that $P_i + \Sigma_j = cI$ or $P_i + \Sigma_j = W$ for the m x m scenario. Although it has already been proven that bias is an insignificant contributor when $P_i + \Sigma_j = cI$ for this scenario it may be desirable to calculate it. The general solution for bias, shown in Equation (6), can be manipulated as follows for case 2:

$$\begin{aligned}
\begin{bmatrix} b_u \\ b_v \end{bmatrix} &= \left(\sum_i^m\sum_j^m W^{-1} d_{ij}\right)^{-1}\left(\sum_i^m\sum_j^m W^{-1}\begin{bmatrix} x_i-u_j \\ y_i-v_j \end{bmatrix} d_{ij}\right) \\
&= \left(\sum_i^m\sum_j^m d_{ij}\right)^{-1} WW^{-1}\left(\sum_i^m\sum_j^m\begin{bmatrix} x_i-u_j \\ y_i-v_j \end{bmatrix} d_{ij}\right) \\
&= \frac{1}{m}\begin{bmatrix} \sum_i^m x_i \sum_j^m d_{ij} - \sum_j^m u_j \sum_i^m d_{ij} \\ \sum_i^m y_i \sum_j^m d_{ij} - \sum_j^m v_j \sum_i^m d_{ij} \end{bmatrix} .
\end{aligned} \tag{7}$$

Equation (7) may be manipulated by employing the constraints identified in Equation (2), such that the equations for the optimal bias are:

$$b_u = \frac{\sum_i^m x_i - \sum_j^m u_j}{m} \text{ and } b_v = \frac{\sum_i^m y_i - \sum_j^m v_j}{m} . \tag{8}$$

Bias is independent of the decision variables and straightforward to compute using Equation (8); therefore, bias may be directly calculated when $P_i + \Sigma_j = cI$ or $P_i + \Sigma_j = W$ for the m x m scenario. This solution is the well-known centroid method for bias removal. Bias is the difference in the positional centroids for the objects of each sensor. This method is optimal if the given assumptions hold true; however, this method may be significantly suboptimal if these conditions do not hold true, especially the assumption that each sensor tracks the same m objects.

3.4 CASE 3: PROCESS POSSIBLE COMBINATIONS

The problem increases in difficulty when the m x n scenario is considered. In the m x n scenario, one sensor tracks m objects and the other tracks n objects, where the m objects are a subset of the n objects. As there are an unequal number of objects, bias is a significant contributor to the overall assignment, even when $P_i + \Sigma_j = cI$ is assumed. The general solution for bias, shown in Equation (6), can be manipulated as follows for case 3:

$$\begin{aligned}
\begin{bmatrix} b_u \\ b_v \end{bmatrix} &= \left(\sum_i^m \sum_j^n W^{-1} d_{ij} \right)^{-1} \left(\sum_i^m \sum_j^n W^{-1} \begin{bmatrix} x_i - u_j \\ y_i - v_j \end{bmatrix} d_{ij} \right) \\
&= \left(\sum_i^m \sum_j^n d_{ij} \right)^{-1} W W^{-1} \left(\sum_i^m \sum_j^n \begin{bmatrix} x_i - u_j \\ y_i - v_j \end{bmatrix} d_{ij} \right) \\
&= \frac{1}{m} \begin{bmatrix} \sum_i^m x_i \sum_j^n d_{ij} - \sum_j^n u_j \sum_i^m d_{ij} \\ \sum_i^m y_i \sum_j^n d_{ij} - \sum_j^n v_j \sum_i^m d_{ij} \end{bmatrix} .
\end{aligned} \tag{9}$$

The algorithm proposed for case 3 is used to calculate the bias and cost of the optimal assignment for a given combination of m objects. The idea behind this method is that if one sensor tracks two objects and the other tracks three, three candidate matches can be searched for the optimal match, as there are *n!/m!(n-m)!* possible ways of choosing m out of n objects. Choosing objects 1 and 2 means that object 3 from the second sensor would not be assigned to one of the objects from the first sensor. The constraints for the $m \times n$ case are defined in Figure 7.

Let S_k, $k = 1{:}\dfrac{n!}{m!(n-m)!}$, be a set of integers $\{j_1 < j_2 < \cdots < j_m \le n\}$ where $n \ge m$.
For each k, the following constraints are:

$$\sum_i d_{ij} = 1, \ \sum_{j \notin S_k} d_{ij} = 0, \text{ and } \sum_i \sum_j d_{ij} = 1$$

Figure 7. Constraints for the m X n scenario.

The constraints shown in Figure 7 are used in conjunction with Equation (9) to derive the following equations for bias:

$$b_u = \frac{\sum_i x_i - \sum_{j \in S_k} u_j}{m} \text{ and } b_v = \frac{\sum_i y_i - \sum_{j \in S_k} v_j}{m} . \tag{10}$$

Therefore, no direct equation exists for bias that is independent of the decision variables for the $m \times n$ scenario when $P_i + \Sigma_j = cI$ or $P_i + \Sigma_j = W$. The bias that is calculated for a given combination is used as input into the objective function, and a track-to-track assignment is calculated for that combination. This procedure is performed for each possible combination, and the assignment with the smallest cost is chosen as the optimal assignment.

3.5 CASE 4: BIAS REMOVAL WHEN COVARIANCES ARE DISSIMILAR

The algorithm for case 4 addresses the bias calculation problem for the $m \times m$ scenario and the $m \times n$ scenario when $P_i + \Sigma_j = W_{ij}$ such that W_{ij} depends on i and j. Direct equations (8) and (10) for bias, were derived from Equation (6) however, when $P_i + \Sigma_j = W$, the decision variables cannot be removed from Equation (6), and there is no explicit equation for calculating bias independently of decision variables. To handle this case, another loop is added to the algorithm such that each possible assignment of m objects is processed, and for each assignment the bias is calculated using Equation (6). For each iteration, the optimal track-to-track assignment is determined using the calculated bias. The overall optimal assignment is then the solution with the smallest cost.

As there generally will be more possible than feasible assignments to process, adaptations to this algorithm can significantly decrease the number of iterations. One technique is to use the measurement error covariance matrix to determine whether an assignment is feasible. The $m \times m$ scenario has only a few feasible assignments where all objects from one sensor are assigned to the objects from the second sensor and are within a given bound of each other. If an error bound on the magnitude of the bias is available, it can also be used to determine feasible assignments. An assignment is assumed to be credible if before bias removal the objects from the two sensors are within a given threshold of each other such that

$$\begin{bmatrix} x_i - u_j \\ y_i - v_j \end{bmatrix}^T \left(P_i + \Sigma_j + P_b\right)^{-1} \begin{bmatrix} x_i - u_j \\ y_i - v_j \end{bmatrix} \leq \lambda^2 \ . \tag{11}$$

An example of these bounds is given in Figure 8, where A, B, and C are the objects from one sensor, and 1, 2, 3, and 4 are the objects from another sensor.

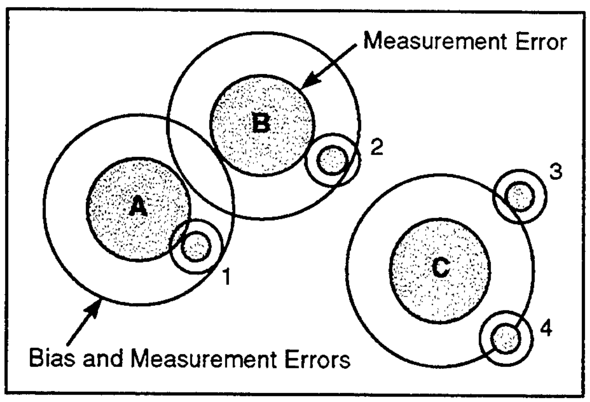

Figure 8. Elimination of possible combinations using the bias error.

A Boolean combination matrix may be constructed that identifies all possible initial combination options. [See Equation (12) for the example in Figure 8.] Note that a value of one means the assignment is feasible; zero means it is infeasible.

$$C=\begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&1&1\end{bmatrix}. \tag{12}$$

This combination matrix reduces the number of iterations from 24 to 2. The idea is to pattern match for objects that can be reasonably assigned; for example, there would be no need to try to match objects A and 3 from Figure 8.

Another adaptation is to assume that one specific object will always be detected and discriminated by sensor one and detected by sensor two, therefore assuming that the object will definitely be assigned from each sensor-object set. Another assumption that improves the accuracy of the calculated bias and decreases computational complexity is that each sensor detects and discriminates one object unambiguously; bias can easily be calculated as the difference between the positions of these two objects. Explanation and development of further adaptations will be presented as future work.

3.6 CASE 5: COMMENTS ON THE MISMATCH SCENARIO

Calculating bias and solving for optimal decision variables is a complex task when the two sensors track different objects. An example of the mismatch scenario is one sensor tracking five objects and another sensor tracking four, but only three of the objects are common. The number and identity of common objects is unknown a priori, as well as the correct value of the bias. The track-to-track assignment problem for the mismatch scenario is nontrivial because of this list of unknowns. Without making assumptions, however, this is the real-time scenario; therefore, algorithms must be robust to operate in the mismatch scenario.

The basic approach to solving the problem proposed here expands on the algorithm presented in case 3 for solving the m x n scenario when $P_i + \Sigma_j = W$. In the m x n scenario, m objects are chosen from the n objects and used to calculate a bias and a track-to-track assignment. The assignment with the smallest cost is chosen as optimal. A similar approach is followed for this case, except now the number of possible combinations is significantly large. It is no longer known that there will be m matched objects, so their number will be varied from 1 to m, with all possible combinations processed.

In this multiple-objective assignment problem, the aims are to (1) calculate bias and find decision variables to minimize the total distance of all assigned objects and (2) maximize the total number of assigned objects. Two possible solution algorithms are discussed. In the first the two goals are weighted to create the new objective function shown in Figure 9. Unfortunately, the weights are problem-dependent parameters chosen through experience. If the only goal were to minimize the distance, the solution would always be to assign only one object such that $b_u = x_i - u_j$, $b_v = y_i - v_j$, and $z = 0$. If the only goal were to maximize the number of assigned objects, m objects would always be assigned, which does not necessarily give the correct pattern match.

The second algorithm assumes that each cost z_k is a chi-square-distributed random variable with n_k degrees of freedom (ndof), such that ndof = 2 x # objects assigned. Because each solution may have different degrees of freedom, each solution must be converted to have a common degree of freedom in order to compare costs. An algorithm to map the cost z_k with n_k degrees of freedom to z_k' with one degree of freedom is shown in Figure 10. The costs for each possible assignment are adjusted for the number of objects assigned, and the solution is simply chosen as the minimal z_k'. This algorithm does not highly encourage solutions where many objects are assigned. To more strongly encourage the second objective, extra weighting may be applied, similar to the first algorithm.

Objective: maximize $v = \sum_k (c_1 q_k - c_2 z_k) d_k$

subject to: $d_k = [0 \text{ or } 1] \quad \forall\, k$

$$\sum_k d_k = 1 \quad \forall\, k$$

where q_k is the number of objects assigned for the kth iteration,
z_k is the optimal cost of the kth iteration after bias removal,
c_1 is the weight of the goal to maximize the number of objects assigned,
c_2 is the weight of the goal to minimize the cost of the assignment, and
d_k is the decision to use the solution of the kth iteration.

Figure 9. Multiple-objective assignment problem.

Let x be a χ^2 distributed random variable with n degrees of freedom.
Let x' be a χ^2 distributed random variable with one degree of freedom.
Let f be the cumulative distribution function that maps a χ^2 random variable with n degrees of freedom to a cumulative probability, p, such that $p = f(x,n)$.
Let g be a function that is the inverse function of f, such that $x = g(p,n) = g(f(x,n),n)$.
Therefore, x is mapped to x' such that $x' = g(f(x,n),1)$.

Figure 10. Algorithm to map a cost to one degree of freedom.

The proposed algorithms for the mismatch scenario are not necessarily practical for a large number of objects, as the number of iterations increases rapidly. For example, if sensor one tracks three objects and sensor two tracks three objects, there are nine ways to assign one object, nine ways to assign two objects, and one way to assign three objects, which totals 19 iterations. Total iterations for a problem with m and n tracked objects are:

$$\sum_{k=1}^{\min(m,n)} \left(\frac{m!}{k!(m-k)!} \right) \left(\frac{n!}{k!(n-k)!} \right) .$$

The number of iterations discussed in this section are for processing the possible combinations when $P_i + \Sigma_j = W$. When $P_i + \Sigma_j = W_{ij}$ significantly more iterations must be processed in addition to each possible assignment discussed in this section. The adaptations discussed in Section 3.5 are equally applicable to decreasing the number of iterations for the mismatch scenario.

3.7 SUMMARY OF CASES

The algorithms described in cases 2 through 4 are related as shown in Figure 11. For case 2, the $m \times m$ scenario where $P_i + \Sigma_j = cI$, bias may be directly calculated using Equation (8). Case 3 is similar to case 2 except that the $m \times m$ scenario is assumed, so bias is calculated by iterating through Equation (10) for all possible combinations of objects. For case 4 an extra loop is added to the flowchart in Figure 11, as all possible assignments are processed using Equation (6). In the mismatch scenario, case 5, Equation (7) is cycled through to calculate the bias for $P_i + \Sigma_j = W$, and Equation (6) is used for $P_i + \Sigma_j = W_{ij}$. As the number of iterations grows rapidly in the mismatch scenario, even for a small number of objects, it is necessary to use techniques to identify the feasible iterations to reduce computer run time.

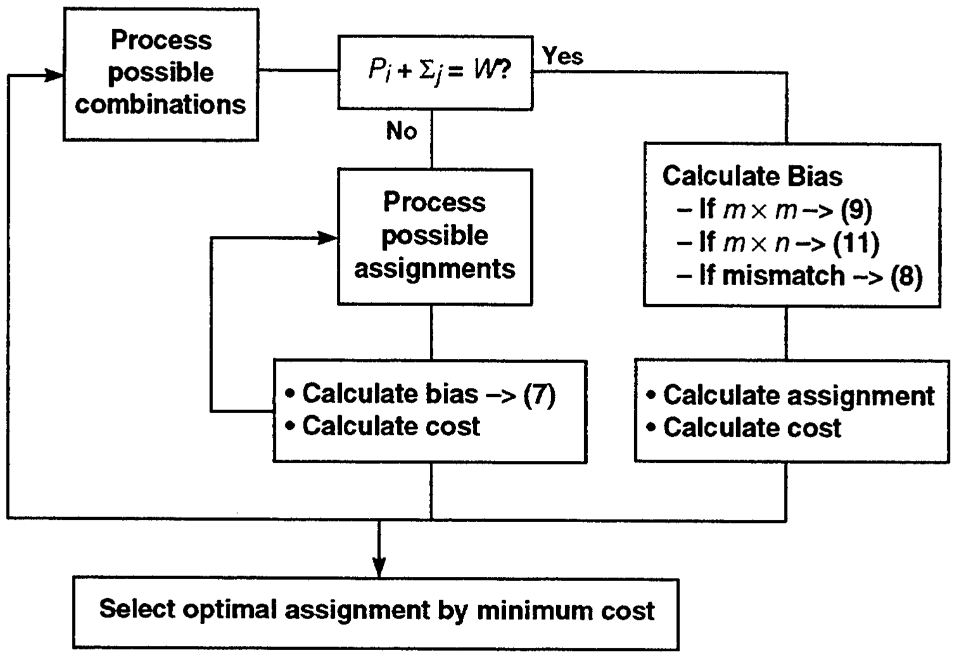

Figure 11. Flowchart of algorithms for cases.

So far this report has emphasized calculating bias as an unknown parameter; however, after bias is calculated it can no longer be modeled in the cost function as in Figure 5. With a given mean $(\hat{b}_u, \hat{b}_v)$ and an error covariance P_b, bias is modeled as a Gaussian random variable. The new cost function to use in solving for the decision variables, (d_{ij}), is shown in Figure 12.

$$c_i = \ln(|P_i|) + \ln(|P_b|) + \ln(|\Sigma_j|) + \begin{bmatrix} x_i - \hat{b}_u - u_j \\ y_i - \hat{b}_v - v_j \end{bmatrix}^T (P_i + P_b + \Sigma_j)^{-1} \begin{bmatrix} x_i - \hat{b}_u - u_j \\ y_i - \hat{b}_v - v_j \end{bmatrix}$$

where $\begin{bmatrix} \hat{b}_u & \hat{b}_v \end{bmatrix}$ is the bias estimate and P_b is the error on the bias estimate

Figure 12. Cost function using estimated bias.

4. CONCLUSIONS

Accurately calculating and removing bias is pertinent to the accurate association of objects between two sensors. This report presents five algorithms for calculating bias. The algorithms cover three scenarios and three assumptions that are based on the number of similar objects and measurement errors, respectively, for the radar and seeker. In a few simple cases bias was shown to be negligible or easily calculated. It was proven that for case 1 bias is an insignificant contributor to the optimal assignment of objects. Two equations were derived to calculate directly the optimal bias for cases 2 and 3. For cases 4 and 5 it was shown that bias is more difficult to calculate than for other special cases and may require an iterative algorithm. Advanced adaptations of the algorithms were briefly discussed; they can drastically help reduce the number of required iterations in many cases. Robust algorithms that use these adaptations will be studied further and implemented as future work.

APPENDIX
DERIVATION OF WEIGHTED-DISTANCE COST FUNCTION

The cost function in Figure 2 is replaced in Figure 5 by a standard weighted-distance cost function. Its derivation is an adaptation of a derivation of the weighted-distance cost function that assumes $P_i + \Sigma_j = W$ [6]. The objective is to find the maximum-probability hypothesis H_{ij} that object i from sensor one is associated with object j from sensor two. The maximum-likelihood estimate of the true position of the object is $\hat{z}$; x_i is the estimate from sensor one; and u_j is the estimate from sensor two. This derivation is for a one-dimensional distance instead of the standard two-dimensional (azimuth and elevation) to simplify the mathematics.

$$\begin{aligned} &\text{maximize } p_{ij} = P(H_{ij}) \\ &p_{ij} = p(x_i|\hat{z})\,p(u_j|\hat{z}) \end{aligned} \qquad \text{(A-1)}$$

Maximizers of p_{ij} also maximize any monotone function of p_{ij}. As the natural logarithm is a monotone function, the maximization problem of Equation (A.1) may be restated as follows:

$$\text{maximize } p_{ij} \Leftrightarrow \text{ minimize } c_{ij} = -\ln(p_{ij}) \;. \qquad \text{(A-2)}$$

Assuming a Gaussian distribution function for the position of the object in Equation (A.3), the following simplifications of Equations (A.4) through (A.7) may be performed:

$$p_{ij} = \frac{1}{2\pi\sqrt{|P_i||\Sigma_j|}} e^{-\frac{1}{2}\left[(x_i-\hat{z})^T P_i^{-1}(x_i-\hat{z}) + (u_j-\hat{z})^T \Sigma_j^{-1}(u_j-\hat{z})\right]} \;, \qquad \text{(A-3)}$$

$$\hat{z} = \left(P_i^{-1} + \Sigma_j^{-1}\right)^{-1}\left(P_i^{-1}x_i + \Sigma_j^{-1}u_j\right) \ , \tag{A-4}$$

$$\begin{aligned} \left(x_i - \hat{z}\right) &= P_i\left(P_i + \Sigma_j\right)^{-1}\left(x_i - u_j\right) \\ \left(u_j - \hat{z}\right) &= -\Sigma_j\left(P_i + \Sigma_j\right)^{-1}\left(x_i - u_j\right) \end{aligned} \ , \tag{A-5}$$

$$\begin{aligned} \left(x_i - \hat{z}\right)^T P_i^{-1}\left(x_i - \hat{z}\right) &= \left(x_i - u_j\right)^T\left(P_i + \Sigma_j\right)^{-1} P_i\left(P_i + \Sigma_j\right)^{-1}\left(x_i - u_j\right) \\ \left(u_j - \hat{z}\right)^T \Sigma_j^{-1}\left(u_j - \hat{z}\right) &= \left(x_i - u_j\right)^T\left(P_i + \Sigma_j\right)^{-1} \Sigma_j\left(P_i + \Sigma_j\right)^{-1}\left(x_i - u_j\right) \end{aligned} \ , \tag{A-6}$$

$$\left(x_i - \hat{z}\right)^T P_i^{-1}\left(x_i - \hat{z}\right) + \left(u_j - \hat{z}\right)^T \Sigma_j^{-1}\left(u_j - \hat{z}\right) = \left(x_i - u_j\right)^T\left(P_i + \Sigma_j\right)^{-1}\left(x_i - u_j\right) \ . \tag{A-7}$$

Using Equation (A.7), (A.3) becomes

$$P_{ij} = \frac{1}{2\pi\sqrt{|P_i||\Sigma_j|}} e^{-\frac{1}{2}\left[\left(x_i - u_j\right)^T\left(P_i + \Sigma_j\right)^{-1}\left(x_i - u_j\right)\right]} \ . \tag{A-8}$$

The weighted-distance cost function is therefore

$$c_{ij} = \ln\left(|P_i|\right) + \ln\left(|\Sigma_j|\right) + \left(x_i - u_j\right)^T\left(P_i + \Sigma_j\right)^{-1}\left(x_i - u_j\right) \tag{A-9}$$

for the one-dimensional position case. The two-dimensional position equivalent of (A.9) is shown in Figure 5.

REFERENCES

1. J. Munkres, "Algorithms for the Assignment and Transportation Problems," *J. SIAM*, 5, 32–38 (1957).

2. H.W. Kuhn, "The Hungarian Method for the Assignment Problem," *Naval Research Logistics Quarterly*, 2, 83–97 (1955).

3. O.E. Drummond, D.A. Castanon, and M.S. Bellovin, "Comparison of 2-D Assignment Algorithms for Sparse, Rectangular, Floating Point Cost Matrices," *SDI Panels on Tracking*, 4, 81–97 (1990).

4. R. Jonker and A. Volgenant, "A Shortest Augmenting Path Algorithm for Dense and Sparse Linear Assignment Problems," *Computing*, 38, 325–340 (1987).

5. F. Hillier and G. Liberman, *Introduction to Operations Research*, New York: McGraw-Hill (1990).

6. K.A. Rink and D.A. O'Connor, "Use of the Auction Algorithm for Target Object Mapping," MIT Lincoln Laboratory, Lexington, Mass., Technical Report TR-1044 (9 February 1998). DTIC ADA-338143.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (*Leave blank*)	2. REPORT DATE 9 July 2002	3. REPORT TYPE AND DATES COVERED Technical Report

4. TITLE AND SUBTITLE

Bias Removal Techniques for the Target-Object Mapping Problem

5. FUNDING NUMBERS

C—F19628-00-C-0002

6. AUTHOR(S)

C.J. Humke

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Lincoln Laboratory, MIT
244 Wood Street
Lexington, MA 02420-9108

8. PERFORMING ORGANIZATION REPORT NUMBER

TR-1060

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Missile Defense Agency
7100 Defense Pentagon
Washington, D.C. 20301

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

ESC-TR-99-057

11. SUPPLEMENTARY NOTES

None

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT (*Maximum 200 words*)

Multisensor track and data fusion require accurate sensor registration and positional bias removal. This report presents the problem of bias removal in the context of midcourse interceptor engagements. For this application two sensors are employed: a fire-control radar and an interceptor with an optical seeker. The fire-control radar tracks objects, collects data on the objects, performs discrimination, and hands over a target object map to the interceptor. Positional bias is generally evident in the target object map and must be removed in order to fuse the radar and seeker data accurately. Equations and algorithms are derived for calculating the maximum-likelihood bias estimate for three scenarios, given three assumptions about the measurement errors on the radar- and seeker-tracked objects. First, bias is shown to be an insignificant contributor to the assignment of objects when both sensors track the same objects and the measurement errors are equal and uncorrelated for all objects. Second, equations for direct calculation of the optimal bias are derived for cases where one sensor tracks a subset of the objects tracked by the other sensor. Third, algorithms are discussed for the most complicated scenarios, those where dissimilar objects are tracked by the two sensors.

14. SUBJECT TERMS	15. NUMBER OF PAGES 40
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	Same as Report